Programmer's Learning Machine

Matthieu Nicolas

IJD Seminar, 2016-02-02

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - Architecture
2. User's code's assessment
   - Problematic
   - The judges
   - Isolation and virtualization
3. Result
4. Next steps

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - Architecture
2. User's code's assessment
   - Problematic
   - The judges
   - Isolation and virtualization
3. Result
4. Next steps

Presentation of PLM

Purposes

- Application to learn programming.

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.

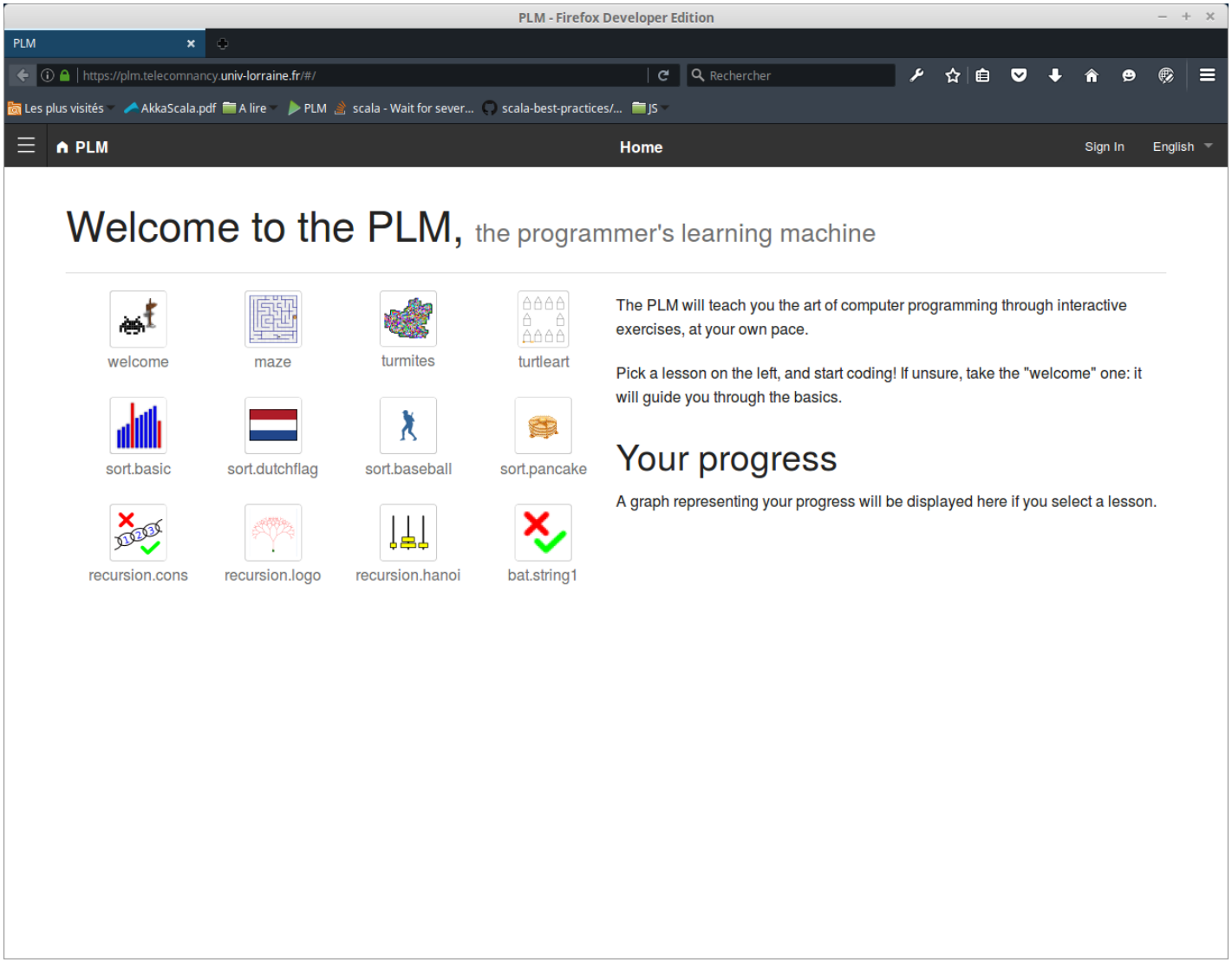
Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.
- Used at TELECOM Nancy since 2008.

Presentation of PLM

Quick demo



Presentation of PLM

Application's architecture

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - Architecture
2. User's code's assessment
   - Problematic
   - The judges
   - Isolation and virtualization
3. Result
4. Next steps

User's code's assessment

Execution components

User's code's assessment

Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops

User's code's assessment

Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files

User's code's assessment

Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?

User's code's assessment

Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?

- Need also to scale.

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - Architecture
2. User's code's assessment
   - Problematic
   - The judges
   - Isolation and virtualization
3. Result
4. Next steps

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...
- ... but user experience needs to be improved!

Result

Live-session in TELECOM Nancy

- Can't cope with the workload.

Result

Live-session in TELECOM Nancy

- Can't cope with the workload.
- No tools for monitoring set up...

Result

Live-session in TELECOM Nancy

- Can't cope with the workload.
- No tools for monitoring set up...
- ... so the bottleneck is unknown.

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - Architecture
2. User's code's assessment
   - Problematic
   - The judges
   - Isolation and virtualization
3. Result
4. Next steps

Next steps

Refactor the code

- Rushed to release a stable version before September...
- Needed to refactor some parts of the code.
- Standardized behavior of local and server mode.

Next steps

Simplify workflow to adapt the content

- Store most content inside PLM.
- Heavy and error prone workflow.
- Need to extract the content from PLM's jar.
- Allow to implement an exercise editor.

Next steps

Solve performance issues

- Set up some monitoring tools.
- Perform some load testing to identify the bottleneck.

Questions

Thanks for your attention, any questions?